**Installing Helm**

Text

Description automatically generated

**Charts**

As described in the [Charts Guide](https://helm.sh/docs/topics/charts), Helm charts are structured like this:

The templates/ directory is for template files. When Helm evaluates a chart, it will send all of the files in the templates/ directory through the template rendering engine. It then collects the results of those templates and sends them on to Kubernetes.

**A Starter Chart**

For this guide, we'll create a simple chart called mychart, and then we'll create some templates inside of the chart.

**$** helm create mychart

Creating mychart

Text

Description automatically generatedGraphical user interface, application

Description automatically generated

### A Quick Glimpse of mychart/templates/

If you take a look at the mychart/templates/ directory, you'll notice a few files already there.

NOTES.txt: The "help text" for your chart. This will be displayed to your users when they run helm install.

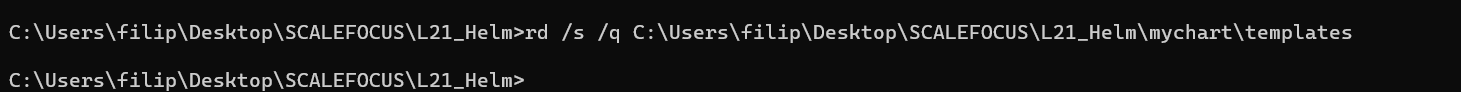
deployment.yaml: A basic manifest for creating a Kubernetes [deployment](https://kubernetes.io/docs/user-guide/deployments/)

service.yaml: A basic manifest for creating a [service endpoint](https://kubernetes.io/docs/user-guide/services/) for your deployment

\_helpers.tpl: A place to put template helpers that you can re-use throughout the chart

And what we're going to do is... remove them all! That way we can work through our tutorial from scratch. We'll actually create our own NOTES.txt and \_helpers.tpl as we go.

**$** rm -rf mychart/templates/



## A First Template

The first template we are going to create will be a ConfigMap. In Kubernetes, a ConfigMap is simply an object for storing configuration data. Other things, like pods, can access the data in a ConfigMap.

Because ConfigMaps are basic resources, they make a great starting point for us.

Let's begin by creating a file called mychart/templates/configmap.yaml:

**apiVersion**: v1

**kind**: ConfigMap

**metadata**:

**name**: mychart-configmap

**data**:

**myvalue**: "Hello World"

**TIP:** Template names do not follow a rigid naming pattern. However, we recommend using the suffix .yaml for YAML files and .tpl for helpers.

The YAML file above is a bare-bones ConfigMap, having the minimal necessary fields. In virtue of the fact that this file is in the mychart/templates/ directory, it will be sent through the template engine.

It is just fine to put a plain YAML file like this in the mychart/templates/ directory. When Helm reads this template, it will simply send it to Kubernetes as-is.

With this simple template, we now have an installable chart. And we can install it like this:

**$** helm install full-coral ./mychart

NAME: full-coral

LAST DEPLOYED: Tue Nov 1 17:36:01 2016

NAMESPACE: default

STATUS: DEPLOYED

REVISION: 1

TEST SUITE: None

Using Helm, we can retrieve the release and see the actual template that was loaded.

Text

Description automatically generatedText

Description automatically generated